Artificial Intelligence-Driven Malware Detection Framework For Internet of Things Environment

Shtwai Alsubai [Corresp., Equal first author, 1], **Ashit Kumar Dutta** [Corresp., Equal first author, 2], **Abdullah Alnajim** [3], **Abdul rahaman Wahab sait** [4], **Rashid Ayub** [5], **Afnan AlShehri** [6], **Naved Ahmad** [6]

[1] Prince Sattam bin Abdulaziz University, Al-Kharj, Kingdom of Saudi Arabia

[2] Department of computer science and information technology, Almaarefa University, Riyadh, Kingdom of Saudi Arabia

[3] Department of Information Technology, College of computer, Qassim University, Buraydah, Saudi Arabia

[4] King Faisal University, Hofuf, Saudi Arabia

[5] King Saud University, Riyadh, Saudi Arabia

[6] Almaarefa University, Riyadh, Kingdom of Saudi Arabia

Corresponding Authors: Shtwai Alsubai, Ashit Kumar Dutta
Email address: Sa.alsubai@psau.edu.sa, adotta@mcst.edu.sa

The Internet of Things (IoT) environment demands a malware detection (MD) framework for protecting sensitive data from unauthorized access.. The study intends to develop an image-based MD framework. The authors apply image conversion and enhancement techniques to convert malware binaries into RGB images. You only look once (Yolo V7) is employed for extracting the key features from the malware images. Harris Hawks optimization is used to optimize the DenseNet161 model to classify images into malware and benign. IoT malware and Virusshare datasets are utilized to evaluate the proposed framework's performance. The outcome reveals that the proposed framework outperforms the current MD framework. The framework generates the outcome at an accuracy and F1-score of 98.65 and 98.5 and 97.3 and 96.63 for IoT malware and Virusshare datasets, respectively. In addition, it achieves an area under the Receiver operating characteristics and the precision-recall curve of 0.98 and 0.85 and 0.97 and 0.84 for IoT malware and Virusshare datasets, accordingly. The study's outcome reveals that the proposed framework can be deployed in the IoT environment to protect the resources.

1 Artificial Intelligence-Driven Malware Detection
2 Framework for Internet of Things Environment

4 Shtwai Alsubai[1], Ashit Kumar Dutta[2], Abdullah M. Alnajim[3], Abdul Rahaman Wahab Sait[4],

5 Rashid Ayub[5], Afnan Mushabbab AlShehri[2], Naved Ahmad[2]

7 [1] Department of Computer Science, College of Computer Engineering and Sciences in Al-Kharj,

8 Prince Sattam bin Abdulaziz University, Al-Kharj, Kingdom of Saudi Arabia.

9 [2] Department of Computer Science and Information Systems, College of Applied Sciences,

10 AlMaarefa University, Ad Diriyah, Riyadh, 13713, Kingdom of Saudi Arabia.

11 [3] Department of Information Technology, College of Computer, Qassim university, Buraydah

12 51452, Kingdom of Saudi Arabia.

13 [4] Department of Archives and Communication, King Faisal University, Al Ahsa, Hofuf, 31982,

14 Kingdom of Saudi Arabia.

15 [5] Department of Science Technology & Innovation Unit, King Saud University, P. O. Box-2454,

16 Riyadh-11451, Saudi Arabia.

19 Corresponding Authors:

20 Shtwai Alsubai[1] and Ashit Kumar Dutta[2]

21 Email address: sa.alsubai@psau.edu.sa and adotta@mcst.edu.sa

23 Abstract

24 The Internet of Things (IoT) environment demands a malware detection (MD) framework for
25 protecting sensitive data from unauthorized access.. The study intends to develop an image-based
26 MD framework. The authors apply image conversion and enhancement techniques to convert
27 malware binaries into RGB images. You only look once (Yolo V7) is employed for extracting the
28 key features from the malware images. Harris Hawks optimization is used to optimize the
29 DenseNet161 model to classify images into malware and benign. IoT malware and Virusshare
30 datasets are utilized to evaluate the proposed framework's performance. The outcome reveals that
31 the proposed framework outperforms the current MD framework. The framework generates the
32 outcome at an accuracy and F1-score of 98.65 and 98.5 and 97.3 and 96.63 for IoT malware and
33 Virusshare datasets, respectively. In addition, it achieves an area under the Receiver operating
34 characteristics and the precision-recall curve of 0.98 and 0.85 and 0.97 and 0.84 for IoT malware
35 and Virusshare datasets, accordingly. The study's outcome reveals that the proposed framework
36 can be deployed in the IoT environment to protect the resources.

37 Introduction

38 The Internet of Things (IoT) connects the real and virtual worlds (Mu et al., 2021; Atitallah, Driss
39 & Almomani, 2022). New business models and global interactions emerge as people, products,

40   technologies, and the internet become more interconnected   (Kumar, Janet & Neelakantan, 2022).
41   Cybercriminals are increasingly targeting IoT devices because they are easy targets for exploiting
42   weak authentication, outdated firmware, and malware due to the complexity of design and
43   implementation in hardware and software (Khan & Salah, 2018). The vulnerabilities in IoT
44   applications can damage the entire network. In recent years, the exponential growth of ML
45   techniques in identifying and categorizing IoT malware effectively. IoT devices are considered
46   heterogeneous, can be implemented on various platforms, and have a wide range of requirements
47   (Li et al., 2022). IoT is subject to the risk of being attacked by malware due to the lack of a practical
48   malware detection (MD) framework. Malware presents a significant threat to IoT security, and one
49   of the crucial issues is detecting unknown malware. Before attempting to implement security
50   solutions, it is essential to understand that IoT devices have significant constraints, such as limited
51   battery retention and low computational processing capability (Meira et al., 2022). Research on
52   IoT security has drawn considerable attention from academic institutions and government
53   agencies. Several studies have identified cyber threats and proposed countermeasures (Emil Selvan
54   et al., 2022).
55   Conventional security approaches have proven ineffective and have failed to deliver decentralized,
56   robust security for IoT networks (Lan et al., 2022). Transitioning similar solutions from traditional
57   platforms to IoT may not be cost-effective due to the limitations of IoT devices (Kan et al., 2021).
58   In addition, IoT platforms allow the integration of network resources into devices not originally
59   conceived of as part of computer networks. Consequently, employing conventional security
60   methods is insufficient to provide IoT systems with MD capabilities (Khan et al., 2020). Due to
61   limited computing resources, conventional MD approaches failed to provide adequate security to
62   IoT devices (Asam et al., 2021). The more reliable and promising performance of machine learning
63   (ML) methods has led to their widespread adoption for MD. With ML algorithms, anti-malware
64   tools have become more effective (Carrillo-Mondéjar, Martínez & Suarez-Tangil, 2020). In recent
65   times, software define networks (SDN) offers an interactive architecture for exchanging the
66   information (Javeed, Gao & Khan, 2021; Javeed et al., 2021; Al Razib et al., 2022). It consists of
67   three layers including, application, control and data. It facilitates a wide range of security for the
68   IoT devices. The centralized architecture and intelligence enable the developers to build an
69   effective MD framework to identify the malware.
70   In the context of encrypted or compromised malware, static signature-based MD algorithms have
71   failed (Asam et al., 2022). Application programming interface data is the most popular feature for
72   building MD models based on sequential behaviours. In contrast, machine activity data is the most
73   popular feature for depicting malware functioning using continuous behaviours (Vignau et al.,
74   2021). Additionally, API calls and opcodes are often employed when developers attempt to
75   determine the common behaviours among malware groups(Shao, Yuan & Wang, 2021). The
76   behavioural-based method is a potential alternative to overcome the challenges of its counterparts.
77   However, over-reliance on behaviours causes the MD models to misclassify a task close to benign
78   functions or mimic lawful behaviours, resulting in a high false-positive rate.

79 Image-based MD can overcome the challenges and limitations of the existing MD models by
80 minimizing the data loss (Falana et al., 2022). Feature engineering is one of the recent
81 developments in image-based MD. There is a strong correlation between feature extraction and
82 the effectiveness of the MD and classification procedure (Saxe & Berlin, 2016; Fathurrahman,
83 Bejo & Ardiyanto, 2022). The existing shallow neural networks and classical ML models demand
84 a higher training duration due to fewer hidden layers(Lirim Ashiku, 2021). It memorizes the
85 training data and makes it difficult to generalize to a newer environment. Deep learning (DL)
86 methods have become increasingly applicable to identifying and analyzing threats with ever-
87 growing malware datasets. Recent studies focus on employing convolutional neural networks
88 (CNN) to classify malware. Deep CNNs facilitate the development of detection systems based on
89 malware images. It enables the MD framework to identify the crucial features of malware.
90 The features learned at lower layers are strengthened in higher layers. These characteristics support
91 CNNs in producing an effective outcome (Kumar, Janet & Neelakantan, 2022). In addition, the
92 computational cost is minimized by limiting the size of the dataset. The grayscale values ranges
93 from 0 to 255 and gradually shifts between the two extremes of black and white.
94 Furthermore, Grayscale images can be created using malware binary. The properties such as
95 texture, intensity, and wavelet can be retrieved from the resulting images (Liu et al., 2020a).
96 Furthermore, recent studies believe that the RGB image can provide more information for
97 classifying malware images. The primary difficulty of visualization methods is computing the
98 texture similarity of a grayscale image. These methods effectively decrypt obfuscated code.
99 However, they are computationally expensive due to the complexity of extracting texture features
100 from malware images. Large datasets make the feature extraction methods less efficient. Malware
101 is constantly evolving, updating, and producing new versions of itself.
102 Consequently, improving the performance of the MD framework with low hardware configuration
103 and extracting relevant information from raw binary data are the primary motivational factors for
104 this study. The study intends to develop an MD framework using the CNN model. In addition, it
105 applies efficient image enhancement and object detection techniques to improve the proposed
106 framework's performance.
107 For IoT devices, there is a requirement for intrusion detection systems that are vastly improved
108 and highly secured. Traditional machine learning algorithms cannot identify sophisticated cyber
109 breaches because of their static design. DL allows for conducting a more in-depth network data
110 analysis and spotting anomalies. Recent studies reveal the crucial role of DL in processing complex
111 images (Falana et al., 2022). Visualizing malware as a coloured image gives the benefit of
112 differentiating various components of the malware binary (Naeem et al., 2020; Jian et al., 2021;
113 Falana et al., 2022). Malware programmers typically modify a small section of the malware codes
114 to develop a new mutant. Thus, visualizing malware as an image offers the benefit of
115 differentiating different components of the malware binary. An image-based DL-driven detection
116 method is highly scalable, flexible, and cost-effective (Naeem et al., 2020; Jian et al., 2021). It can
117 evaluate vast amounts of data and automatically alter security systems to identify malware or
118 security breaches with minimum processing resources.

119 The contributions of the study are:

120 i. An effective technique to generate images from malware binaries.It overcomes the challenges

121 of the existing RGB image generation technique. In addition, it reduces the possibility of data loss

122 during the image generation process.

123 ii. A feature extraction technique for extracting the key features from the malware images. It

124 provides the critical features for the CNN models. By presenting a set of crucial features, the

125 performance of the MD model is improved.

126 iii. A hybrid CNN model for detecting malware in the IoT environment. It addresses the limitations

127 of the existing MD techniques by employing an image-based detection technique. In addition, it

128 demands a minimum hardware and software configuration compared to the recent CNN models.

129 iv. The proposed model achieved a significant outcome in detecting malware in IoT environment

130 compared to the current models in terms of accuracy, precision, recall, and F1-measure.

131 The remaining part of this study is organized as follows: Section 2 outlines the recent MD using

132 images and binary files. The study's methodology is discussed in section 3. Sections 4 and 5

133 highlight the performance analysis of the proposed framework and compare it with the recent MD

134 frameworks. Finally, section 6 concludes this study.

135 **Literature Review**

136 The field of image processing extensively employed CNN to generate a practical outcome

137 (Smmarwar, Gupta & Kumar, 2022). The weight sharing and the convolution kernel methods were

138 used in CNN to overcome the limitations of neural network techniques (Asam et al., 2021).

139 Recently, researchers have focused on improving the malware visualization technique's

140 performance and reducing computation cost. This section covers visualization-related studies,

141 including malware identification using statistical similarity measures, machine learning, and deep

142 learning. Traditional MD techniques primarily analyse harmful code properties (Conti, Khandhar

143 & Vinod, 2022). These capabilities also utilize advanced machine learning-based MD techniques

144 to identify new forms of destructive code. However, these technologies failed to detect new

145 malware variations.

146 Several malware analysis visualization methods have been suggested recently. Makandar & Patrot,

147 (2017) developed a novel approach for detecting malware using image features. They generated

148 two-dimensional grayscale graphics from the structure of the compressed binary executable. Based

149 on the findings, binary texture analysis proved more precise and efficient. Venkatraman, Alazab

150 & Vinayakumar, (2019) proposed an image based model for detecting malware. Vasan et al.,

151 (2020) proposed an approach for converting raw binaries into colour images and detecting malware

152 families. They employed data augmentation for processing the imbalanced dataset. Malimg

153 malware and IoT-android mobile datasets were used for performance evaluation. The outcome

154 shows that the model can identify hidden code and malware families with limited resources.

155 Liu et al., (2020) introduced a reinforcement method that relies on ML to identify various forms

156 of malware and its variations. Naeem et al., (2020) developed an MD method for the Industrial

157 Internet of Things (IIoT). To track and record information about incoming and outgoing traffic,

158 the authors developed a sniffer gateway. Awan et al. (2021) introduced an image–based malware

159 classification. They employed the VGG-19 network to classify 25 well-known malware images.
160 Jian et al., (2021) suggested a unique deep neural network–based visual MD methodology. They
161 established that three-channel RGB images are superior to grayscale images for malware
162 identification.
163 Similarly, Sharma et al., (2022) proposed an Xception CNN-based MD framework for classifying
164 malware images. The authors stated that the models achieve a superior outcome than the current
165 frameworks. Yadav et al., (2022) developed a MD framework using Andriod malware images.
166 Obaidat et al., (2022) proposed a CNN model using Java bytecode. Chaganti, Ravi & Pham, (2022)
167 developed a Bi-directional DL approach for classifying the IoT malware images. Bensaoud &
168 Kalita, (2022) employed Malimg dataset for evaluating the CNN model. In another study, Falana
169 et al., (2022) developed a technique to convert malware binaries into an image to support the
170 process of malware classification. A slight variation in an image assists CNN models in identifying
171 critical malware. They employed three benchmark datasets: MaleVis, Mallmg, and Virusshare.
172 The findings suggested that the model achieves an average accuracy of 96.77%. The recent
173 techniques focussed on pattern-based MD. However, it has many drawbacks, including a high false
174 positive rate that causes many valid activities to be incorrectly labelled intrusive. There is a demand
175 for more critical training data. In addition, the existing methods require high-end computation
176 resources to generate an effective outcome. Table 1 outlines the features of the existing MD
177 frameworks.
178   Table 1 Related Works
179

Materials and Methods

181 The authors propose a DL based framework to classify malware and benign images based on the
182 study's objective. Figure 1 highlights the three phases of the proposed framework. In phase 1, the
183 authors convert the binaries into an image. The images are pre-processed and resized as 600 x 600
184 pixels. The authors employ You look only once (Yolo) V7 to identify critical features from the
185 images. Phase 2 involves Harris Hawks optimization (HHO) to fine-tune the DenseNet161
186 parameters to identify malware from the datasets. Finally, phase 3 evaluates the performance of
187 the proposed frameworks. The authors utilize two datasets in this study, including IoT malware
188 and binaries. Two IoT malware datasets (IoT_malware and Virusshare) are used in this study
189 which is available in the Elmasry dataset (Malware, 2021) and Virusshare dataset (Virusshare,
190 2021) respectively. IoT_malware dataset is a recently developed malware images dataset. It
191 includes the IoT malware images of categories including benign and malware. The unpacked
192 executable and linkage format binaries for malware and benign applications were represented in
193 the image format. In addition, the Virusshare dataset contains instances of multiple malware
194 families. The description of the datasets is provided in table 2.
195   Table 2 Dataset characteristics
196
197   Figure 1: Proposed MD Framework
198

199 Based on the SDN framework, the researchers framed the network model as shown in figure 2 for
200 implementing the proposed model. In the control layer, the binaries are converted into images and
201 transformed as RGB images. Yolo V7 extracts the crucial objects. Finally, the fined tuned CNN
202 model classifies the malware and benign images.
203   Figure 2: Proposed Network Model
204 In phase 1, the authors follow the approaches of Falana et al., (2022) to convert binaries into an
205 image. Let B_1, B_2, …, B_n and M_1, M_2, …, M_n be the benign and malware binaries set, respectively.
206 Let D be a space to hold benign and malware binaries. Therefore, D_i represents a binary, which
207 may be benign or malicious. Figure 3 shows converting binaries and grayscale images into RGB
208 images.
209
210   Figure 3: RGB Image Conversion Processes
211
212 The following algorithm presents the algorithm for transforming the binaries (D) into a grayscale
213 image.
214 Algorithm: Grayscale to RGB image
215 Input: D
216 Output: Grayscale Image (I)
217 Bit_array = 0; // 8 bit array
218 Pix_arr[0][0] = 0; // A pixel array
219 I_width = 2^8
220 I_height = sizeof (D) / 2^{11}
221 for i = 1 to len (D) – 1 do
222   for c = 1 to F
223   if c = 0 or c=1
224   Bit_array = c
225   if I % 8 ==0
226   Continue
227   endif
228   endif
229   endfor
230   for k = 1 to 255
231   X[k] = Bit_array[k]
232   for l = 1 to len(Bit_array)
233   Pix_arr[k][l] = Bit_array[k]
234   endfor
235   endfor
236 endfor
237 return pix_arr
238 During the image pre-processing phase, the grayscale images (G) are converted into RGB images
239 (RGB). Initially, the luminosity method converts a grayscale image into an RGB image. Eqn. 1
240 represents the conversion process of G into RGB.

241   $RGB (I) = G[(0.3 * R) + (0.59 * G) + (0.11 * B)]$ (1)

242   Insufficient or non-uniform RGB images contain a significant amount of noise. Therefore, the
243   authors enhance the RGB image based on (Mu et al., 2021). Eqn. 2 shows the intensity
244   enhancement of RGB(x,y).

245   $RGB(x,y) = a_k RGB(x,y) + \overline{b_k}$ (2)

246   where a_k and b_k are linear coefficients.

247   Eqns. 3,4, and 5 outline the process of brightness equalization using adaptive gamma correction.

248   $RGB(x,y) = RGB(x,y)^{\emptyset(x,y)}$ (3)

249   where $\emptyset(x,y)$ is the gamma correction.

250   $\emptyset(x,y) = \frac{RGB(x,y) + \propto}{1 + \propto}$ (4)

251   $\propto = 1 - \frac{1}{cd}\Sigma_{x=1}^{c}\Sigma_{y=1}^{d}RGB(x,y)$ (5)

252   where $\propto$ is the adaptively derived mean value of $G(x,y)$, and c,d is the height and width of RGB
253   (x,y).

254   In addition, image fusion filters noises from the RGB image. Eqns. 6, 7, and 8 show the denoised
255   reflection component (DNI) and enhanced RGB image.

256   $DNI = \overline{a_i}RGB(x,y) + \overline{b_i}$ (6)

257   $Enhanced\ RGB(x,y) = \frac{1}{1 + \exp(-8(DNI - b))}$ (7)

258   $b = \frac{1}{cd}\Sigma_{x=1}^{c}\Sigma_{y=1}^{d}RGB(x,y)$ (8)

259   where b is image intensity, and c and d are the images' height and width.

260   In this phase, the RGB images are enhanced in order to assist the DenseNet model. The high –
261   standard deviation in the images are adjusted to reduce the variation in the pixel value. Image
262   fusion filter is applied to remove the noises by blurring the images. It adjusts the uneven pixel
263   value and removes the chromatic aberration. In the subsequent step, Canny edge detection is
264   employed for identifying the ranges of edges. The non-maximum suppression is used to thin out
265   the edges. The intensity of the images are identified using Double threshold method. Finally,
266   contrast limited adaptive histogram equalization is employed to improve the image quality.

267   Furthermore, the authors employ Yolo V7 (Wang, Bochkovskiy & Liao, 2022) to extract
268   meaningful features from the images. Yolo V7 achieves a superior outcome with fewer
269   computational resources. It generates an output faster without any pre-trained weights. It uses CNN
270   for extracting features and predicting the probability of classes. Yolo V7 overcome the challenges
271   in its previous versions. It contains an extended efficient layer aggregation network and compound
272   model scaling technique that support the proposed model for detecting malware images. Yolo V7
273   architecture includes residual blocks, bounding box and intersection over union (IoU). It divides
274   the images into multiple grids (residual blocks) with equal dimensions. Each grid is a region to
275   highlight the object. It consists of width (w), height (h), class (c), and center (x,y)

276

277   Figure 4: Bounding box region

278   Eqn. 9 represents the bounding box that highlights a region in figure 4.

279 $$Y = \left(P_c, bb_x, bb_y, bb_h, bb_w, C\right) \tag{9}$$

280 where P_c is the probability of an object in the bounding box (bb).

281 IoU is a metric for evaluating the performance of Yolo V7. It measures the Yolo V7's ability to

282 detect the malware dataset's features. Eqn. 10 shows the expression of IoU.

283 $$IoU = \frac{Area\ of\ overlapping\ of\ actual\ and\ predicted\ malware\ feature}{Area\ of\ union\ of\ actual\ and\ predicted\ malware\ feature} \tag{10}$$

284 Yolo V7 computes the IoU score for each object detection process. An IoU score greater than 0.5

285 represents the better performance of an object detection model.

286 In phase 2, the authors optimize the DenseNet161 model using the HHO algorithm. HHO

287 algorithm is one of the recent optimization algorithms for improving the performance of the

288 complex models. It tunes the CNN model's parameters for improving the classification accuracy.

289 It minimizes the error rate and searches for the optimal learning rate for identifying the malware

290 and benign images. The architecture of DenseNet161 comprises an activation function, a pooling

291 layer, a dropout layer, and the convolutional layer. Each layer acquires information from the

292 previous layer and guides the subsequent layers. DenseNet161 simplifies the connecting pattern

293 among the layers. It reuses malware and benign image features and enhances the network's

294 performance. In addition, it requires a limited number of parameters compared to its counterparts.

295 The developmental rate controls the number of data in a layer. Each dense block includes two

296 convolutions, and each dense layer contains two operations to extract malware and benign features

297 and reduces its depth. HHO is a familiar swarm-based optimization technique. It is used to improve

298 the performance of the DenseNet161 model. In the context of MD, HHO identifies the effective

299 parameters (number of pooling layers, dropout layer, and convolutional layers) for generating the

300 outcome.

301 The malware and benign images are considered a rabbit in the HHO searching environment. The

302 HHO searching strategies support the proposed framework to classify the images. The exploration

303 and exploitation phases assist the DenseNet161 model in identifying the malware's exact location

304 and benign images. Let q be the equal chance between the DenseNet161 parameters. The HHO

305 exploitation phase for the proposed framework is modelled in Eqn. 11.

306 $$M(t + 1) = \begin{cases} \left(M_{rand}(t) - Y_1 \mid M_{rand}(t) - 2Y_2 M(t)\right), q \geq 0.5 \\ M_{malben}(t) - M_m(t) - Y_3\left(LB + Y_4(UB - LB)\right), q < 0.5 \end{cases} \tag{11}$$

307 Where M(t+1) is the location of the DenseNet161 parameters in the subsequent iteration, M_{malben}

308 (t) is the location of malware and benign image. M(t) is the present position of hawks, Y1, Y2, Y3,

309 Y4, and q are arbitrary numbers between 0 and 1, frequently modified at each iteration. LB and

310 UB are the lower and upper bounds of each variable, M_{rand}(t)is an arbitrary DenseNet161 parameter

311 from the present population, and M_m is the average location of the parameter. Eqns. 12 and 13

312 represent the soft besiege of malware and benign images in the HHO environment.

313 $$M(t + 1) = \Delta M(t) - E \left| J M_{malben}(t) - M(t) \right| \tag{12}$$

314 $$\Delta M(t) = M_{malben}(t) - M(t) \tag{13}$$

315 where $\Delta M(t)$ is the difference between the malware and benign image and the present position in
316 iteration t, E is the parameter to represent the transition between soft and hard besiege, and J is the
317 jump strength. Hard besiege is described in Eqn. 14.

318 $\quad M(t+1) = M_{malben}(t) - E|\Delta M(t)|$ $\hspace{5cm}$ (14)

319 The following algorithm outlines the HHO algorithm for optimizing the Densenet161 parameters.

320

321 Algorithm: HHO pseudocode for DenseNet161
322 Input : Dataset D and number of iteration (epoch) T
323 output: Malware image classification and its fitness value
324 Initialize the population at random M_i (i=1, 2,…, D)
325 While (d in D) do
326 $\quad$ Calculate the fitness value (parameters) of DenseNet161
327 $\quad$ Set M_{malben} as the location of malware and benign images
328 For each parameter (M_i) do
329 Update E_0 and J // E_0 = 2 rand () -1 and J = 2(1-rand ())
330 $\left(\frac{T-t}{T}\right)E_0 2$ $\quad$ update E using $\quad$ E =
331 $\quad$ if $(|E| \geq 1)$ then
332 $\quad\quad$ Update the location using Eqn. 11
333 $\quad$ if $(|E| < 1)$ then
334 $\quad\quad$ if$(r \geq 0.5\,\&|E| \geq 0.5)$ then
335 $\quad\quad\quad$ Update the location using Eqn. 12
336 $\quad\quad$ else if $(r \geq 0.5\,\&|E| < 0.5)$ then
337 $\quad\quad\quad$ update the location using Eqn. 14
338 $\quad$ return M_{malben}

339

340 In phase 3, the authors apply precision, recall, F1-measure, accuracy, Matthews correlation
341 coefficient (MCC), and Kappa to evaluate the proposed framework's performance. The dataset is
342 divided into a train set (70%) and test set (30%). In the MD environment, precision is the number
343 of malware and benign classification among the classified images. A recall is a set of classified
344 malware and benign images. F1-score is the harmonic mean of a number of malware and benign
345 images in the datasets and correctly detected images. Accuracy is the number of optimally
346 classified malware and benign images. MCC is the difference between predicted malware and
347 benign images and actual malware and benign images.
348 Furthermore, it summarizes the confusion and error matrices. Cohen's Kappa compares the
349 classified malware and benign images with the expected accuracy. It addresses the evaluation bias
350 by providing the chances of generating optimal classification using a random guess. In addition,
351 the error rate and computation cost are calculated for each classification.

352 **Results**

353 In this section, the authors highlight the experimental outcome of this study. The proposed model
354 is implemented in Windows 10 professional environment, i7 processor, GTX 1080 Ti (11 GB).
355 Python 3.9 with Keras (Keras, 2022) library is employed for developing the proposed framework,

356   Vasan et al., framework, Jian et al., framework, Sharma et al., framework, and Falana et al.,
357   framework. In addition, YoloV7 (Wang, Bochkovskiy & Liao, 2022), DenseNet161 (DenseNet161,
358   2022)and HHO (HHO, 2022) are utilized for constructing the model.
359   The similar hardware and software configuration is followed for the training phase. During the
360   training phase, the DenseNet161 parameters are supervised by the HHO algorithm.The authors
361   train the DenseNet161 model with IoT datasets under the HHO environment to identify critical
362   parameters for generating an optimal outcome. During the training phase, the proposed MD
363   framework generates an optimal result at the 32nd and 37th epoch for IoT malware and Virusshare
364   datasets, respectively. Furthermore, the authors extended the training to the 37th and 40th epochs
365   for the IoT malware and Virusshare datasets. However, there is no significant improvement in the
366   model's performance. Thus, epoch values and the dropout ratios of 32 and 41, 0.3 and 0.5, are
367   assigned for IoT malware and Virusshare datasets, respectively. Based on the outcome of the
368   hyperparameter optimization, an array of five layers comprised of two fully connected layers, three
369   dropout layers and an activation function are integrated with the DenseNet161 model. The
370   hyperparameter tuning process identifies an optimal set of DenseNet161 parameters to detect
371   malware images from the dataset.
372   Table 3 outlines the performance of the proposed framework. In the testing phase, the trained
373   DenseNet161 model achieves an average accuracy, precision, recall, F1-measure, MCC, and
374   Kappa of 98.65, 98.7, 98.3, 98.5, 97.5, and 97.65, respectively, for the IoT malware dataset. HHO
375   assists the DenseNet161 model in generating optimum results. The outcome reveals the adequate
376   performance of the proposed MD model. The higher value of MCC and Kappa indicates that the
377   proposed model classifies the images with optimal precision on the imbalanced dataset.
378   **Table 3:** Performance analysis for the IoT Malware dataset
379
380   Likewise, Table 4 shows the proposed framework's performance on the Virusshare dataset. Compared to
381   the IoT malware dataset, the Virusshare dataset contains many files. Moreover, the image conversion model
382   supports the proposed framework for converting the binaries into an RGB image. The proposed model
383   achieves an optimal accuracy on the Virusshare dataset. The feature extraction process assists the proposed
384   model in identifying the crucial features of the images. Figures 5(a) and 6(b) illustrate the performance of
385   the proposed framework on the IoT malware and Virushare datasets. It shows that the model effectively
386   classifies the malware and benign images. In addition, the proposed model addresses the overfitting
387   challenges on the IoT malware and the Virusshare datasets.
388   **Table 4:** Performance analysis for the Virusshare dataset
389
390   Figure 5: (a) IoT Malware Dataset (b) Virusshare Dataset
391   Table 5 highlights the comparative analysis's outcome of the MD framework. The proposed
392   framework outperforms the recent MD frameworks. The high value of Kappa suggests the
393   effectiveness of the proposed MD framework on the imbalanced dataset. In addition, it highlights
394   the importance of the proposed MD framework in handling true and false positives. However,
395   Falana et al. framework produces a reasonable outcome on the IoT malware dataset.
396   **Table 5:** Comparative analysis of the IoT Malware dataset
397

398   Likewise, table 6 presents the results of the comparative analysis of the Virusshare dataset. The
399   proposed MD framework obtained a superior MCC and Kappa on the Virusshare dataset. Yolo
400   V7 and HHO algorithm enables the proposed framework to produce a superior outcome. In
401   addition, the image enhancement technique offers the proposed framework to identify the key
402   objects. However, both Falana et al., and Vasan et al., frameworks achieve results similar to the
403   proposed framework.

404   **Table 6:** Comparative analysis of the Virusshare dataset

405

406   Figure 6 reflects the performance of the individual MD frameworks on the IoT malware and the
407   Virusshare datasets, respectively. The proposed feature extraction method offers the key features
408   of the DenseNet161 model to classify malware and benign images effectively.

409

410   Figure 6: Comparative analysis outcome of the MD frameworks
411   Figure 7 highlights the area under the Receiver operating characteristics (AU-ROC) and the MD
412   frameworks' precision-recall curve (AU-PRC) on the IoT malware dataset. The proposed
413   framework achieves AU-ROC and AU-PRC values of 0.98 and 0.85, respectively. On the other
414   hand, the AU-ROC and AU-PRC values of Jian et al. (0.88 and 0.75), Sharma et al. (0.89 and
415   0.77), Falana et al. (0.87 and 0.69) and Vasan et al., (0.84 and 0.74), accordingly. The outcome
416   represents the classification efficiency of the proposed MD. The higher values of AU-ROC and
417   AU-PRC indicates that the MD model detects the features of malware and benign images,
418   effectively.

419

420   Figure 7: AU – ROC and AU-PRC for the IoT malware
421   Likewise, figure 8 displays the AU-ROC and AU-PRC of the MD frameworks. The proposed
422   framework reaches the AU-ROC and AU-PRC of 0.97 and 0.84, respectively. In contrast, the
423   remaining frameworks achieve the AU-ROC and AU-PRC of Jian et al. (0.92 and 0.77), Sharma
424   et al. (0.90 and 0.81), Falana et al. (0.92 and 0.73), and Vasan et al.,(0.78 and 0.76), accordingly.

425

426   Figure 8: AU – ROC and precision and recall curve for dataset_2
427   Table 7 outlines the error rate of MD frameworks. The proposed framework produces fewer errors
428   for IoT malware (14.2%) and Virusshare (15.6%). The feature extraction phase assists the
429   proposed framework in generating a superior outcome compared to the other frameworks.

430   **Table 7:** Error rate of CNN models
431   Finally, Table 8 presents the computational complexities of the MD frameworks in classifying the
432   malware images. Compared to the existing frameworks, the proposed framework consumes fewer
433   parameters, learning rate, and computation time.

434   **Table 8:** Computational complexities of the CNN models

435

Discussion

437   The authors developed an image-based MD framework for identifying malware and benign files
438   in the IoT environment. An image conversion technique converts malware and benign binaries

439 into a grayscale image. Furthermore, the grayscale images are enhanced to RGB images. An object
440 identification technique extracts a key feature from the images. Yolo V7 is a recent CNN technique
441 for identifying the crucial elements of malware and benign images. HHO algorithm is used to
442 optimize the DenseNet161 model for classifying malware and benign images. It identifies the
443 critical parameters of the DenseNet model in order to detect malware within a limited amount of
444 time. DenseNet161 contains a set of hyper-parameters that reinforces the model to find the crucial
445 objects from the images.   Predictive accuracy and detection rates are the primary metrics for
446 evaluating MD frameworks. The primary step in securing a system and gaining control over its
447 further malware spread is accurately discovering the previously undetected instances. Improving
448 the detection accuracy of a proposed method may result in false alarms. Attempts to reduce false
449 alarms may have an unintended negative effect on detection efficiency. As a result, the proposed
450 model uses dissimilarity by contrasting the harmonic mean of both factors, known as the F1
451 measure. In addition, MCC and Kappa are used to measure the efficiency of the proposed
452 framework.
453 The image format enables the MD framework to serve multiple types of platforms. In addition, the
454 CNN model can identify a slight variation in textures and patterns in the images. Thus, the
455 proposed model supports the SDN framework to offer a protective environment for the IoT
456 devices. The study uniquely integrates image enhancement, object detection (Yolo V7), and hyper-
457 parameter tuned CNN model (HHO – DenseNet161). Image enhancement and object detection
458 reduces the computation overhead of the proposed model. The hyperparameter optimization tunes
459 the key parameters such as number of dropout layers and epochs. The fined tuned model classifies
460 the images with limited resources.
461 Jian et al. (2021) developed a deep neural network architecture called SERLA (SEResNet50 +
462 Bidirectional Long Short Term Memory (Bi-LSTM) + attention) to increase the performance of
463 the detection approach. However, the proposed model's performance outperforms the Jian et al.,
464 model due to the effective image enhancement and feature extraction techniques. In addition, the
465 computation cost for constructing Bi-LSTM is higher than the proposed method. The framework
466 of Sharma et al. (2022) generated a better outcome; however, the computation cost is higher than
467 the proposed MD framework. Falana et al. (2022) framework comprised a CNN and generative
468 neural network for classifying the malware images. However, there is a lack of feature engineering
469 or extraction process to identify the critical features from the images. In addition, the complex
470 architecture requires additional computation time to generate the outcome. In line with the Vasan
471 et al. (2020) framework, the recommended MD framework applied the HHO algorithm to fine-
472 tune the DenseNet161. Figure 6 reflects the MD performance on IoT malware and Virusshare
473 datasets. It shows that the proposed MD outperforms the recently developed image-based MD. In
474 line with the studies (Obaidat et al., 2022; Yadav et al., 2022; Smmarwar, Gupta & Kumar, 2022),
475 the proposed model achieves a superior outcome. The significant improvement in the feature
476 extraction and image classification processes enabled the proposed MD to achieve a better
477 outcome. The existing models (Chaganti, Ravi & Pham, 2022; Kumar, Janet & Neelakantan, 2022)
478 generated a reasonable outcome. However, the computation cost was very high comparing to the

479   proposed MD framework. Tables 7 and 8 reveals the error rates and the computational
480   complexities of the MD frameworks. It is evident that the proposed model require less
481   computational resources for detecting the malware. The proposed study's outcome follows the
482   studies of (Vinayakumar et al., 2018; Su et al., 2018; Rabbani et al., 2020; Naeem et al., 2020;
483   Javeed, Gao & Khan, 2021; Javeed et al., 2021; Anand et al., 2021; Awan et al., 2021; Al Razib
484   et al., 2022) for protecting the computational resources from the malware.
485   The presently offered MD technologies are only effective on traditional networks. The
486   implementation of the models are difficult to apply on IoT networks or do not possess the
487   flexibility and robustness necessary to ensure secure operations. The study's outcome reveals that
488   they are appropriate for securing the IoT. It is adaptable, distributed, resilient, and does not require
489   many computational resources. Many IoT devices, including temperature and humidity sensors,
490   used in environmental and agricultural applications are battery-powered and deployed in distant
491   places, necessitating an MD technique that is both computationally and energy-efficient to extend
492   the battery life of these devices. The proposed framework can be applied in environmental and
493   agricultural applications to minimize energy consumption and protect the network. IoT-based
494   systems in smart cities rely on various devices, such as security cameras, that collect personal
495   information and need stringent security protocols to prevent unauthorized access. Safeguarding
496   the IoT system against malware is critical for the well-being of the workforce and the sustained
497   improvement of the Industrial IoT. Thus, the proposed MD framework can offer an effective
498   industrial working environment and safeguard crucial computing resources in industrial settings.
499   The proposed model yields reliable results and aids in identifying malware in IoT networks. In
500   future investigations, several limitations should be addressed. CNN's multiple layers increase
501   training time and demand a GPU. Nevertheless, the current IoT framework facilitates the high end
502   software and hardware configuration for implementing a DL based detection method. In addition,
503   the proposed MD model is a lightweight application comparing to the recent models. Therefore,
504   the proposed MD model can operate in multiple IoT platforms. The study's findings reveal that
505   the proposed MD model require limited computational resources. The hyperparameter tuned CNN
506   model achieved a better outcome. The existing CNN and recurrent neural network approaches
507   failed to present a crucial pattern from the malware binaries due to data loss and irrelevant features.
508   Yolo V7 model assists the proposed MD framework by providing the key features of malware.
509   The proposed image based MD framework overcome the challenges of the existing approaches.
510   The proposed technique may suffer from the imbalanced dataset. The data pre-processing is
511   required to improve an image's quality and deliver high performance. There is a possibility of
512   losing critical features due to multiple features. The inability to use coordinate frames might render
513   the graphics unfavorably. The architecture of the proposed model necessitates a sizable quantity
514   of data to yield an exciting result. However, the researcher introduced image enhancement and
515   feature extraction to handle the shortcomings of the CNN model. Incorporating feature selection
516   results into the images' internal representation can yield positive results.

517   **Conclusion**

518 The authors present the image-based MD framework for the IoT environment in this study. The
519 malware binaries are converted into images to improve the quality of the malware classification
520 approach. In addition, an image enhancement technique is employed to convert the grayscale
521 images to RGB images. An object identification method is used for feature extraction to support
522 the trained convolutional neural network approach. For classifying the malware images, the
523 authors employed the DenseNet161 model with the support of the Harris Hawks optimization
524 algorithm. The performance evaluation was conducted on IoT malware and Virusshare datasets.
525 The experimental outcome shows that the proposed framework is suitable for real-time
526 applications.
527 Moreover, the framework is lightweight, which demands a low computation cost for generating
528 effective results. Thus, the framework can be applied to small and large-scale industries. It
529 performs better on IoT malware and Virusshare datasets. However, there is a demand for additional
530 experimentation to improve the performance of the proposed MD framework. In the future, the
531 authors intend to extend the framework with the generative adversarial network to generalize the
532 proposed framework's implementation to other malware image datasets.

Funding Statement

534 The authors would like to acknowledge the support provided by AlMaarefa University while
535 conducting this research work. The researchers would like to thank the Deanship of Scientific
536 Research Qassim University for funding the publication of this project. This work was supported
537 by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific
538 Research, King Faisal University, Saudi Arabia [Grant No. 2740]

539

References

541 Anand A, Rani S, Anand D, Aljahdali HM, Kerr D. 2021. An Efficient CNN-Based Deep
542 Learning Model to Detect Malware Attacks (CNN-DMA) in 5G-IoT Healthcare
543 Applications. *Sensors 2021, Vol. 21, Page 6346* 21:6346. DOI: 10.3390/S21196346.
544 Asam M, Hussain SJ, Mohatram M, Khan SH, Jamal T, Zafar A, Khan A, Ali MU, Zahoora U.
545 2021. Detection of exceptional malware variants using deep boosted feature spaces and
546 machine learning. *Appl. Sci.* 11:21. DOI: 10.3390/app112110464.
547 Asam M, Khan SH, Akbar A, Bibi S, Jamal T, Khan A, Ghafoor U, Bhutta MR. 2022. IoT
548 malware detection architecture using a novel channel boosted and squeezed CNN. *Scientific
549 Reports 2022 12:1* 12:1–12. DOI: 10.1038/s41598-022-18936-9.
550 Atitallah S Ben, Driss M, Almomani I. 2022. A Novel Detection and Multi-Classification
551 Approach for IoT-Malware Using Random Forest Voting of Fine-Tuning Convolutional
552 Neural Networks. *Sensors 2022, Vol. 22, Page 4302* 22:4302. DOI: 10.3390/S22114302.
553 Awan MJ, Masood OA, Mohammed MA, Yasin A, Zain AM, Damaševičius R, Abdulkareem
554 KH, Sauveron D. 2021. Image-Based Malware Classification Using VGG19 Network and
555 Spatial Convolutional Attention. *Electronics 2021, Vol. 10, Page 2444* 10:2444. DOI:
556 10.3390/ELECTRONICS10192444.
557 Bensaoud A, Kalita J. 2022. Deep multi-task learning for malware image classification. *Journal
558 of Information Security and Applications* 64:103057. DOI: 10.1016/J.JISA.2021.103057.
559 Carrillo-Mondéjar J, Martínez JL, Suarez-Tangil G. 2020. Characterizing Linux-based malware:
560 Findings and recent trends. *Futur. Gen. Comput. Syst.* 110:267–281. DOI:

561   10.1016/j.future.2020.04.031.

562 Chaganti R, Ravi V, Pham TD. 2022. Deep learning based cross architecture internet of things
563   malware detection and classification. *Computers & Security* 120:102779. DOI:
564   10.1016/J.COSE.2022.102779.

565 Conti M, Khandhar S, Vinod P. 2022. A few-shot malware classification approach for unknown
566   family recognition using malware feature visualization. *Computers and Security* 122. DOI:
567   10.1016/j.cose.2022.102887.

568 DenseNet161 G. 2022.DenseNet161. *Available at https://github.com/flyyufelix /DenseNet-*
569   *Keras/blob/master/densenet161.py* (accessed October 10, 2022).

570 Emil Selvan GS, Azees M, Rayala Vinodkumar CH, Parthasarathy G. 2022. Hybrid optimization
571   enabled deep learning technique for multi-level intrusion detection. *Advances in*
572   *Engineering Software* 173. DOI: 10.1016/j.advengsoft.2022.103197.

573 Falana OJ, Sodiya AS, Onashoga SA, Badmus BS. 2022. Mal-Detect: An intelligent
574   visualization approach for malware detection. *Journal of King Saud University - Computer*
575   *and Information Sciences* 34:1968–1983. DOI: 10.1016/J.JKSUCI.2022.02.026.

576 Fathurrahman A, Bejo A, Ardiyanto I. 2022. Lightweight Convolution Neural Network for
577   Image-Based Malware Classification on Embedded Systems. *2021 International Seminar*
578   *on Machine Learning, Optimization, and Data Science, ISMODE 2021*:12–16. DOI:
579   10.1109/ISMODE53584.2022.9743111.

580 HHO G. 2022.Harris Hawk Optmization. *Available at https://github.com/cahitberkay/Harris-*
581   *Hawks-Optimization-HHO---Python-Code/blob/master/HHO.py* (accessed September 25,
582   2022).

583 Javeed D, Gao T, Khan MT. 2021. SDN-Enabled Hybrid DL-Driven Framework for the
584   Detection of Emerging Cyber Threats in IoT. *Electronics 2021, Vol. 10, Page 918* 10:918.
585   DOI: 10.3390/ELECTRONICS10080918.

586 Javeed D, Gao T, Khan MT, Ahmad I. 2021. A Hybrid Deep Learning-Driven SDN Enabled
587   Mechanism for Secure Communication in Internet of Things (IoT). *Sensors 2021, Vol. 21,*
588   *Page 4884* 21:4884. DOI: 10.3390/S21144884.

589 Jian Y, Kuang H, Ren C, Ma Z, Wang H. 2021. A novel framework for image-based malware
590   detection with a deep neural network. *Computers & Security* 109:102400. DOI:
591   10.1016/J.COSE.2021.102400.

592 Kan X, Fan Y, Fang Z, Cao L, Xiong NN, Yang D, Li X. 2021. A novel IoT network intrusion
593   detection approach based on Adaptive Particle Swarm Optimization Convolutional Neural
594   Network. *Information Sciences* 568:147–162. DOI: 10.1016/J.INS.2021.03.060.

595 Keras. 2022.kera library. *Available at https://keras.io* (accessed July 7, 2022).

596 Khan MA, Salah K. 2018. IoT security: Review, blockchain solutions, and open challenges.
597   *Future Generation Computer Systems* 82:395–411. DOI: 10.1016/j.future.2017.11.022.

598 Khan A, Sohail A, Zahoora U, Qureshi AS. 2020. A survey of the recent architectures of deep
599   convolutional neural networks. *Artif. Intell. Rev.* 53:5455–5516. DOI: 10.1007/s10462-020-
600   09825-6.

601 Kumar S, Janet B, Neelakantan S. 2022. Identification of malware families using stacking of
602   textural features and machine learning. *Expert Systems with Applications* 208. DOI:
603   10.1016/j.eswa.2022.118073.

604 Lan J, Liu X, Li B, Sun J, Li B, Zhao J. 2022. MEMBER: A multi-task learning model with
605   hybrid deep features for network intrusion detection. *Computers and Security* 123. DOI:
606   10.1016/j.cose.2022.102919.

607 Li K, Ma W, Duan H, Xie H, ZHU J. 2022. Few-shot IoT attack detection based on RFP-CNN
608   and adversarial unsupervised domain-adaptive regularization. *Computers and Security* 121.
609   DOI: 10.1016/j.cose.2022.102856.
610 Lirim Ashiku CD. 2021. Network intrusion detection system using deep learning. *Procedia*
611   *Computer Science* 185:239–247.
612 Liu X, Lin Y, Li H, Zhang J. 2020a. A novel method for malware detection on ML-based
613   visualization technique. *Computers and Security* 89. DOI: 10.1016/J.COSE.2019.101682.
614 Liu K, Xu S, Xu G, Zhang M, Sun D, Liu H. 2020b. A Review of Android Malware Detection
615   Approaches Based on Machine Learning. *IEEE Access* 8:124579–124607. DOI:
616   10.1109/ACCESS.2020.3006143.
617 Makandar A, Patrot A. 2017. Malware class recognition using image processing techniques.
618   *2017 International Conference on Data Management, Analytics and Innovation, ICDMAI*
619   *2017*:76–80. DOI: 10.1109/ICDMAI.2017.8073489.
620 Malware I. 2021.IoT Malware, Kaggle. *Available at*
621   *https://www.kaggle.com/datasets/anaselmasry/iot-malware* (accessed August 8, 2021).
622 Meira J, Eiras-Franco C, Bolón-Canedo V, Marreiros G, Alonso-Betanzos A. 2022. Fast
623   anomaly detection with locality-sensitive hashing and hyperparameter autotuning.
624   *Information Sciences* 607:1245–1264. DOI: 10.1016/j.ins.2022.06.035.
625 Mu Q, Wang X, Wei Y, Li Z. 2021. Low and non-uniform illumination color image
626   enhancement using weighted guided image filtering. *Computational Visual Media 2021 7:4*
627   7:529–546. DOI: 10.1007/S41095-021-0232-X.
628 Naeem H, Ullah F, Naeem MR, Khalid S, Vasan D, Jabbar S, Saeed S. 2020. Malware detection
629   in industrial internet of things based on hybrid image visualization and deep learning model.
630   *Ad Hoc Networks* 105:102154. DOI: 10.1016/J.ADHOC.2020.102154.
631 Obaidat I, Sridhar M, Pham KM, Phung PH. 2022. Jadeite: A novel image-behavior-based
632   approach for Java malware detection using deep learning. *Computers & Security*
633   113:102547. DOI: 10.1016/J.COSE.2021.102547.
634 Rabbani M, Wang YL, Khoshkangini R, Jelodar H, Zhao R, Hu P. 2020. A hybrid machine
635   learning approach for malicious behaviour detection and recognition in cloud computing.
636   *Journal of Network and Computer Applications* 151:102507. DOI:
637   10.1016/J.JNCA.2019.102507.
638 Al Razib M, Javeed D, Khan MT, Alkanhel R, Muthanna MSA. 2022. Cyber Threats Detection
639   in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework. *IEEE Access*
640   10:53015–53026. DOI: 10.1109/ACCESS.2022.3172304.
641 Saxe J, Berlin K. 2016. Deep neural network based malware detection using two dimensional
642   binary program features. *2015 10th International Conference on Malicious and Unwanted*
643   *Software, MALWARE 2015*:11–20. DOI: 10.1109/MALWARE.2015.7413680.
644 Shao Z, Yuan S, Wang Y. 2021. Adaptive online learning for IoT botnet detection. *Inf. Sci. (Ny)*
645   574:84–95. DOI: 10.1016/j.ins.2021.05.076.
646 Sharma O, Sharma A, Kalia A. 2022. Windows and IoT malware visualization and classification
647   with deep CNN and Xception CNN using Markov images. *Journal of Intelligent*
648   *Information Systems 2022*:1–27. DOI: 10.1007/S10844-022-00734-4.
649 Smmarwar SK, Gupta GP, Kumar S. 2022. Deep malware detection framework for IoT-based
650   smart agriculture. *Computers and Electrical Engineering* 104. DOI:
651   10.1016/j.compeleceng.2022.108410.
652 Su J, Danilo Vasconcellos V, Prasad S, Daniele S, Feng Y, Sakurai K. 2018. Lightweight

653 Classification of IoT Malware Based on Image Recognition. *Proceedings - International*
654 *Computer Software and Applications Conference* 2:664–669. DOI:
655 10.1109/COMPSAC.2018.10315.
656 Vasan D, Alazab M, Wassan S, Naeem H, Safaei B, Zheng Q. 2020. IMCFN: Image-based
657 malware classification using fine-tuned convolutional neural network architecture.
658 *Computer Networks* 171:107138. DOI: 10.1016/J.COMNET.2020.107138.
659 Venkatraman S, Alazab M, Vinayakumar R. 2019. A hybrid deep learning image-based analysis
660 for effective malware detection. *Journal of Information Security and Applications* 47:377–
661 389. DOI: 10.1016/J.JISA.2019.06.006.
662 Vignau B, Khoury R, Hallé S, Hamou-Lhadj A. 2021. The evolution of IoT malwares, from
663 2008 to 2019: Survey, taxonomy, process simulator and perspectives. *J. Syst. Archit.*
664 116:102143. DOI: 10.1016/j.sysarc.2021.102143.
665 Vinayakumar R, Soman KP, Poornachandran P, Sachin Kumar S. 2018. Detecting Android
666 malware using Long Short-term Memory (LSTM). *Journal of Intelligent and Fuzzy Systems*
667 34:1277–1288. DOI: 10.3233/JIFS-169424.
668 Virusshare D. 2021.Virusshare dataset. *Available at https://virusshare.com* (accessed August 8,
669 2021).
670 Wang C-Y, Bochkovskiy A, Liao H-YM. 2022. YOLOv7: Trainable bag-of-freebies sets new
671 state-of-the-art for real-time object detectors. DOI: 10.48550/arxiv.2207.02696.
672 Yadav P, Menon N, Ravi V, Vishvanathan S, Pham TD. 2022. A two-stage deep learning
673 framework for image-based android malware detection and variant classification. *Computational*
674 *Intelligence* 38:1748–1771. DOI: 10.1111/COIN.12532.

Related works

1   **Table 1 Related works**

Authors (year)	Dataset	Method	Results (%)	Platform
Makandar & Patrot, (2017)	Malimg	Support vector machine and Discrete wavelet transform (DWT)	Average accuracy: 98.2	Non- IoT platform
Venkatraman, Alazab & Vinayakumar, (2019)	Malimg	CNN	Average accuracy:98.4	Real-world application
Vasan et al., (2020)	Malimg and IoT Android	Gated recurrent unit and CNN	Achieved accuracy of 98.82 on Malimg and 97.35 on IoT android	IoT platform
Liu et al., (2020)	Microsoft BIG	CNN and GAN	Average accuracy : 96.25	Real-world application
Naeem et al., (2020)	Leopard Mobile dataset	Deep CNN model	Average accuracy: 97.5	Industrial IoT
Awan et al., (2021)	Malimg	Spatial attention CNN	Average accuracy: 97.62	IoT platform
Asam et al., (2021)	Malimg	Deep boosted feature space-based malware classification	Average accuracy:98.6	Real-world application
Sharma, Sharma & Kalia, (2022)	Windows malware binaries	Xception CNN	Average accuracy: 97.5	Real-world application
Smmarwar, Gupta & Kumar, (2022)	IoT malware and Malimg	DWT, GAN and CNN	Average accuracy:99	IoT platform
Conti, Khandhar & Vinod, (2022)	Windows malware binaries	Convolutional Siamese neural network	Average accuracy:98.5	Real-world application
Yadav et al., (2022)	Android malware images	SVM and Random Forest	Achieved an average accuracy of 92.9 on multi-class and 100 on binary class	Android
Kumar & Janet, (2022)	Malimg and	VGG16, VGG19, ResNet50, and	Average accuracy: 98.92	Real-world application

	Microsoft BIG	Google's inception V3		
Obaidat et al., (2022)	Java bytecode	CNN	accuracy: 98.4	Real-world application
Falana et al., (2022)	Malevis, Malimg, and Virusshare	Deep GAN and CNN	Average accuracy: 96.77	Real-world application
Chaganti, Ravi & Pham, (2022)	IoT malware	Bi-directional DL model	Average accuracy: 98	IoT platform
Bensaoud & Kalita, (2022)	Malimg	Deep CNN	Average accuracy: 97.5	Real-world application

Dataset characteristics

1   Table 1 Dataset characteristics

Dataset	Files	Classification	Size (in MB)
IoT_malware	17,186	2	64
Virusshare	30,967	2	73

Performance analysis for the IoT Malware dataset

Table 2: Performance analysis for the IoT Malware dataset

Methods / Measures	Accuracy	Precision	Recall	F1 - Measure	MCC	Kappa
Training						
Malware	98.3	98.1	98.3	98.2	97.8	97.9
Benign	98.6	98.4	98.5	98.45	97.5	98.1
Average	98.45	98.25	98.4	98.33	97.65	98
Testing						
Malware	98.7	98.6	98.1	98.35	97.6	97.5
Benign	98.6	98.8	98.5	98.65	97.4	97.8
Average	98.65	98.7	98.3	98.5	97.5	97.65

Performance analysis for the Virusshare dataset

Table 3: Performance analysis for the Virusshare dataset

Methods / Measures	Accuracy	Precision	Recall	F1 - Measure	MCC	Kappa
Training						
Malware	97.5	97.8	96.3	97.04	95.8	95.4
Benign	97.1	96.5	97.1	96.8	95.7	95.6
Average	97.3	97.15	96.7	96.92	95.75	95.5
Testing						
Malware	97.4	96.1	97.2	96.65	95.2	94.8
Benign	97.2	96.7	96.5	96.6	95.3	94.9
Average	97.3	96.4	96.85	96.63	95.25	94.85

Comparative analysis of the IoT Malware dataset

Table 4: Comparative analysis of the IoT Malware dataset

Methods / Measures	Accuracy	Precision	Recall	F1 - Measure	MCC	Kappa
Proposed framework	98.65	98.7	98.3	98.5	97.5	97.65
Jian et al., framework	97.6	96.8	96.5	96.65	94.3	94.6
Sharma et al., framework	97.8	97.4	97.6	97.5	95.2	96.1
Falana et al., framework	98.2	97.5	97.3	97.4	95.6	96.4
Vasan et al., framework	97.9	98.1	97.5	97.8	96.4	96.1

Comparative analysis of the Virusshare dataset

1   **Table 5:** Comparative analysis of the Virusshare dataset

Methods / Measures	Accuracy	Precision	Recall	F1 - Measure	MCC	Kappa
Proposed framework	97.3	96.4	96.85	96.63	95.25	94.85
Jian et al., framework	95.8	96.2	95.7	95.95	93.1	93.8
Sharma et al., framework	96.2	95.1	96.1	95.6	93.5	94.2
Falana et al., framework	97.1	96.2	95.6	95.9	94.3	93.5
Vasan et al., framework	96.5	95.8	96.3	96.05	94.6	93.7

Error rate of CNN models

Table 6: Error rate of CNN models

Methods / Dataset	IoT Malware Dataset (%)	Virusshare Dataset (%)
Proposed framework	14.2	15.6
Jian et al., framework	14.8	16.4
Sharma et al., framework	15.3	17.1
Falana et al., framework	15.7	16.2
Vasan et al., framework	16.3	15.9

Computational complexities of the CNN models

Table 7: Computational complexities of the CNN models

Methods / Dataset	IoT Malware Dataset			Virusshare Dataset		
	No. of parameters	Learning rate	Testing time (seconds)	No. of parameters	Learning rate	Testing time (seconds)
Proposed framework	6.4 M	1×10^{-4}	152.3	7.1 M	1×10^{-5}	148.9
Jian et al., framework	6.8 M	1×10^{-3}	167.2	8.4 M	1×10^{-4}	153.4
Sharma et al., framework	7.4 M	1×10^{-2}	154.5	8.7 M	1×10^{-3}	161.6
Falana et al., framework	7.3 M	1×10^{-3}	151.6	7.6 M	1×10^{-4}	159.5
Vasan et al., framework	6.5 M	1×10^{-3}	150.5	6.9 M	1×10^{-5}	161.1

Figure 1

proposed_malware_detection_framework
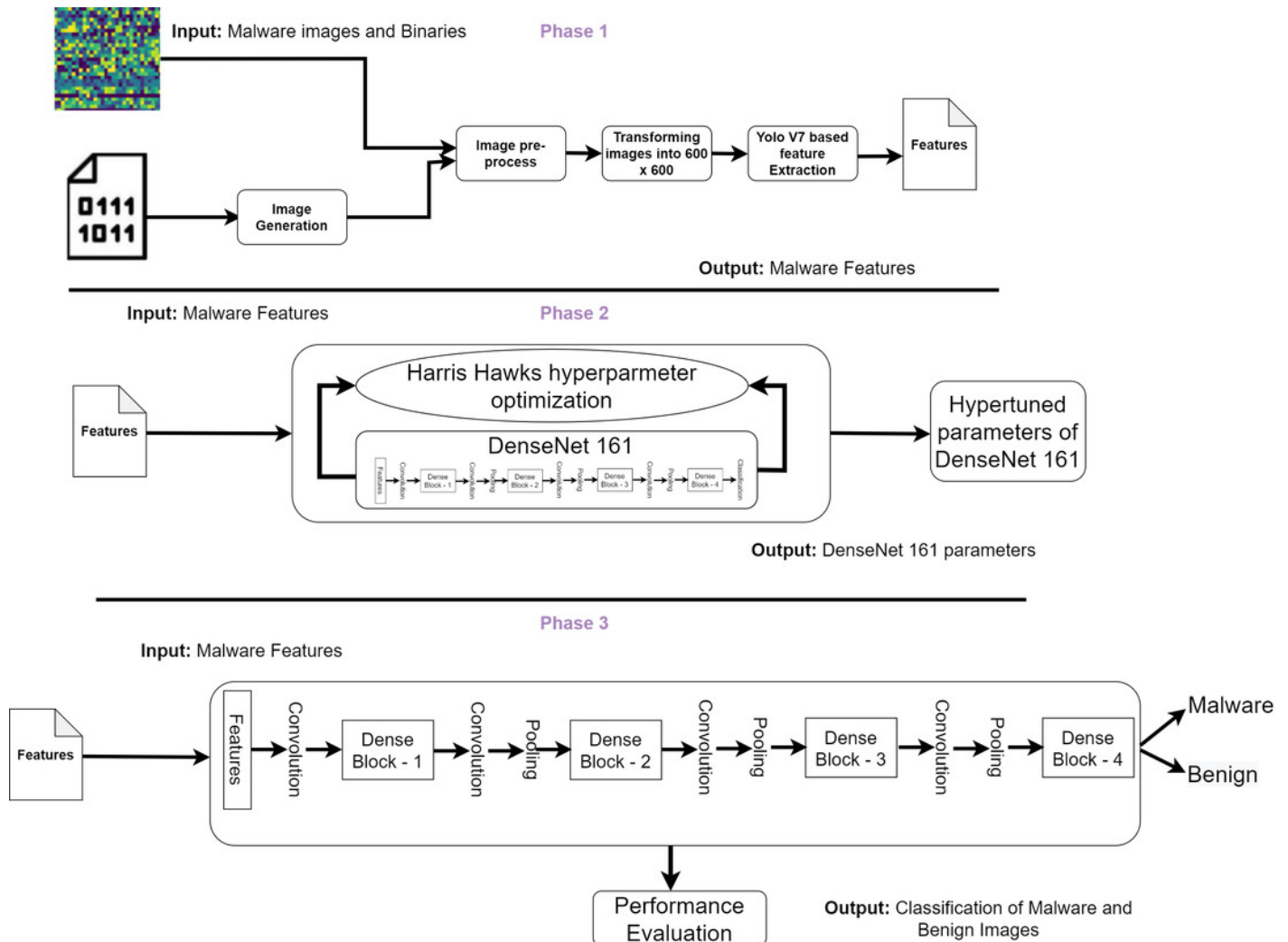
Figure 2

Proposed Network model
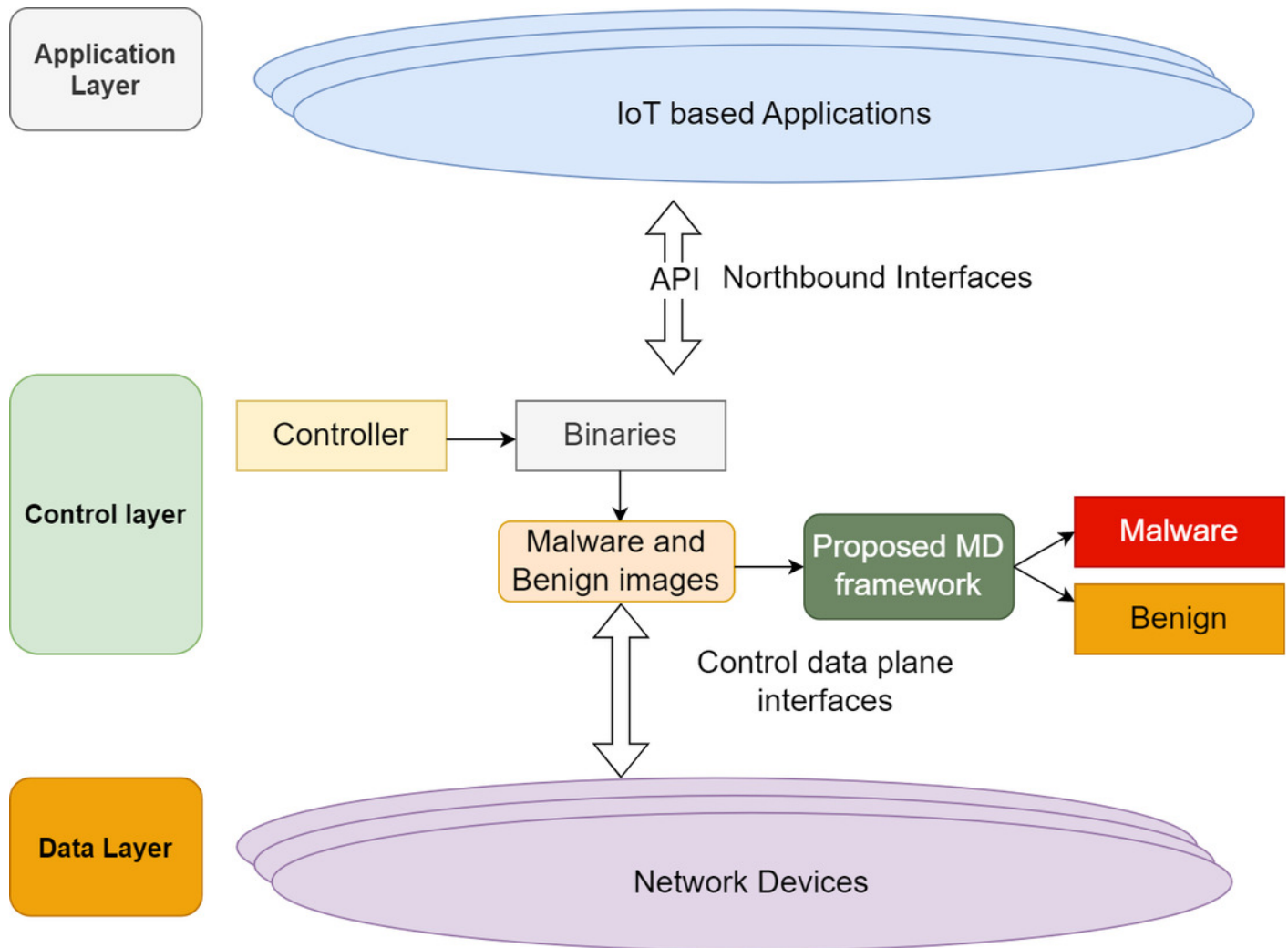
Figure 3

RGB Image Conversion process
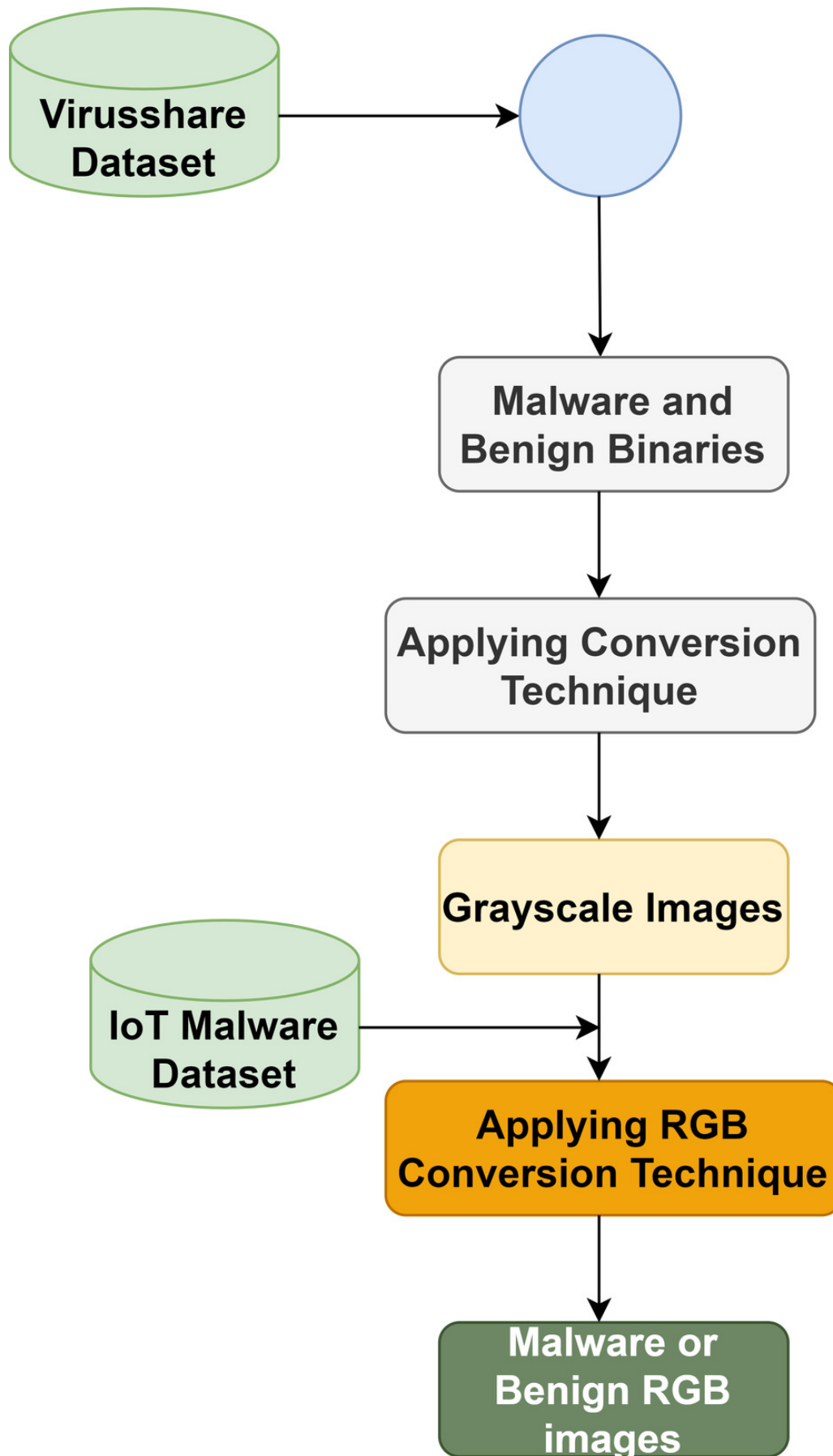
Figure 4

Bounding Box region
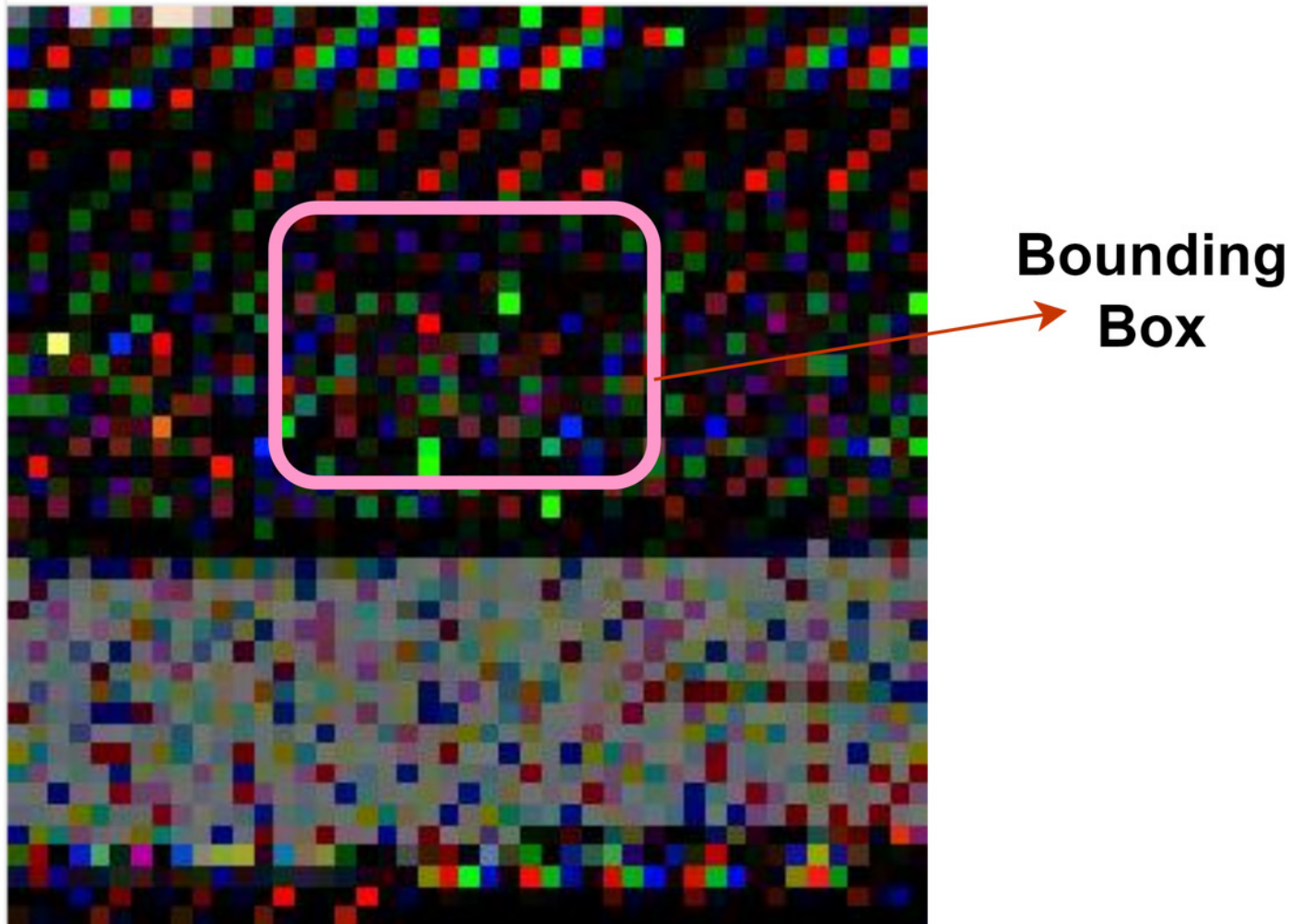
Figure 5

Comparative analysis outcome of the MD frameworks
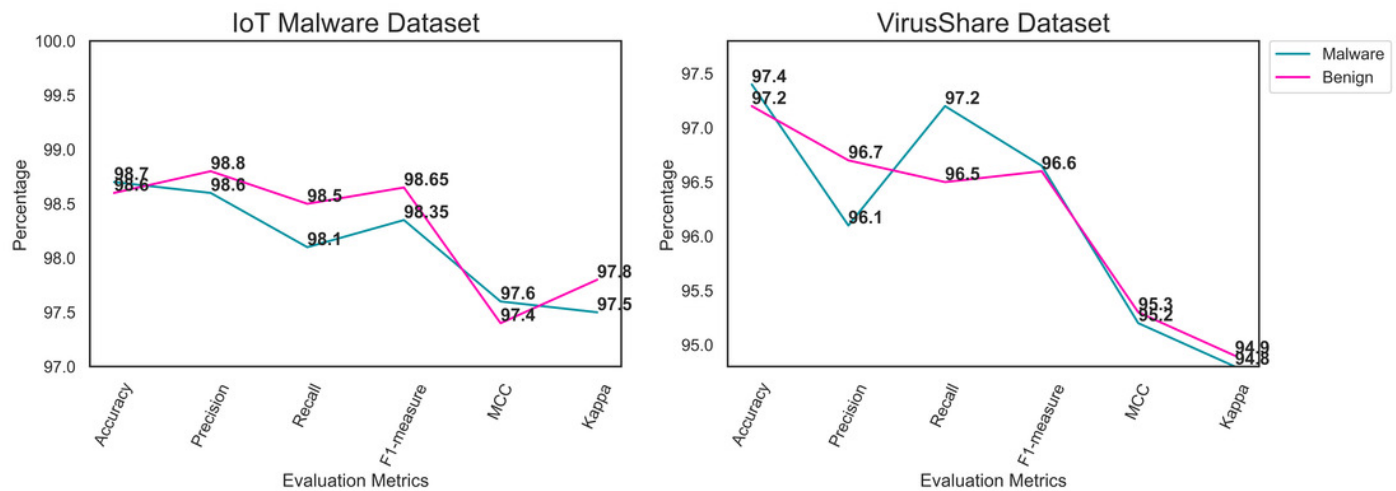
Figure 6

AU – ROC and AU-PRC for the IoT malware
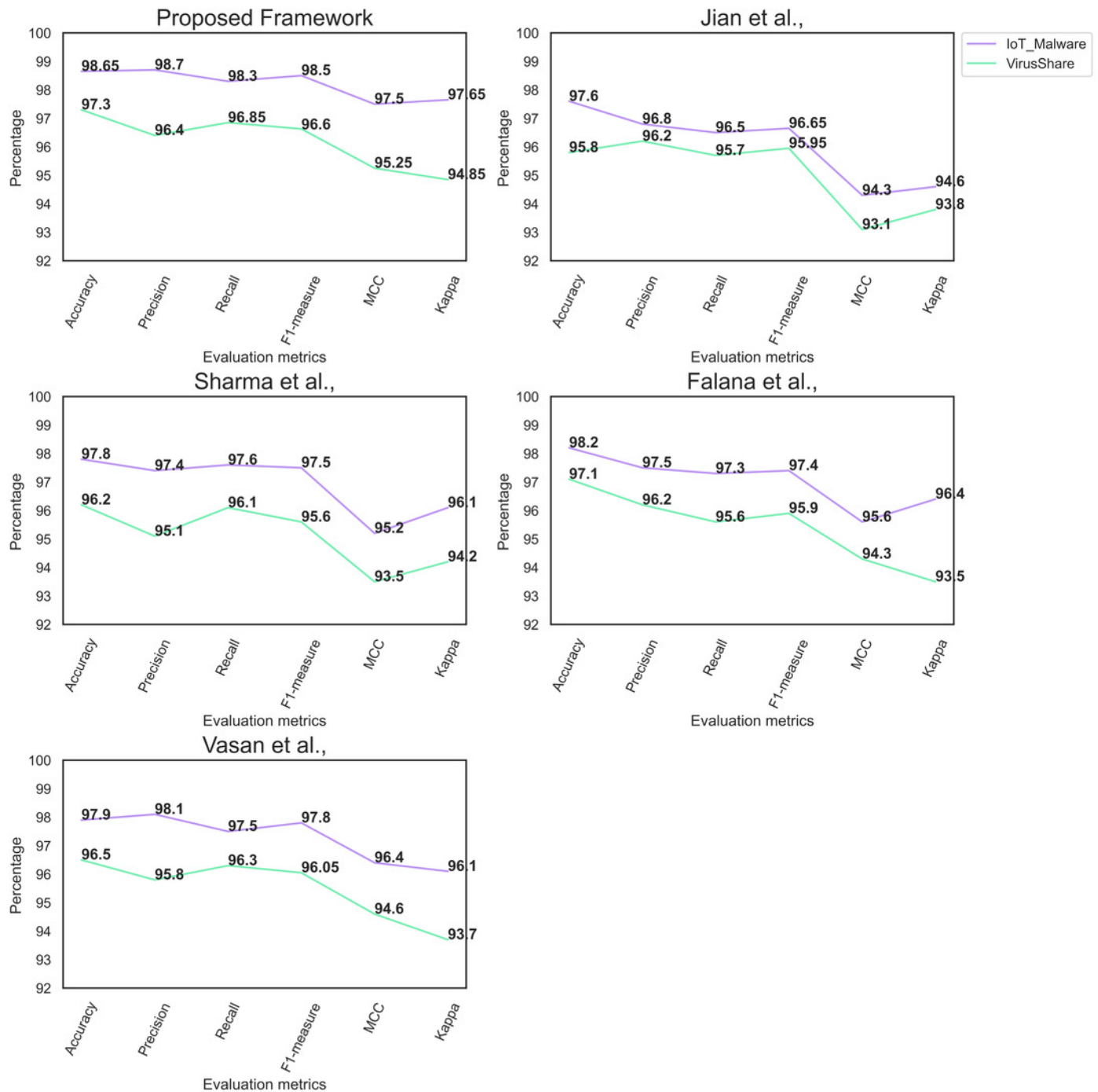
Figure 7

AU – ROC and precision and recall curve for virusshare dataset
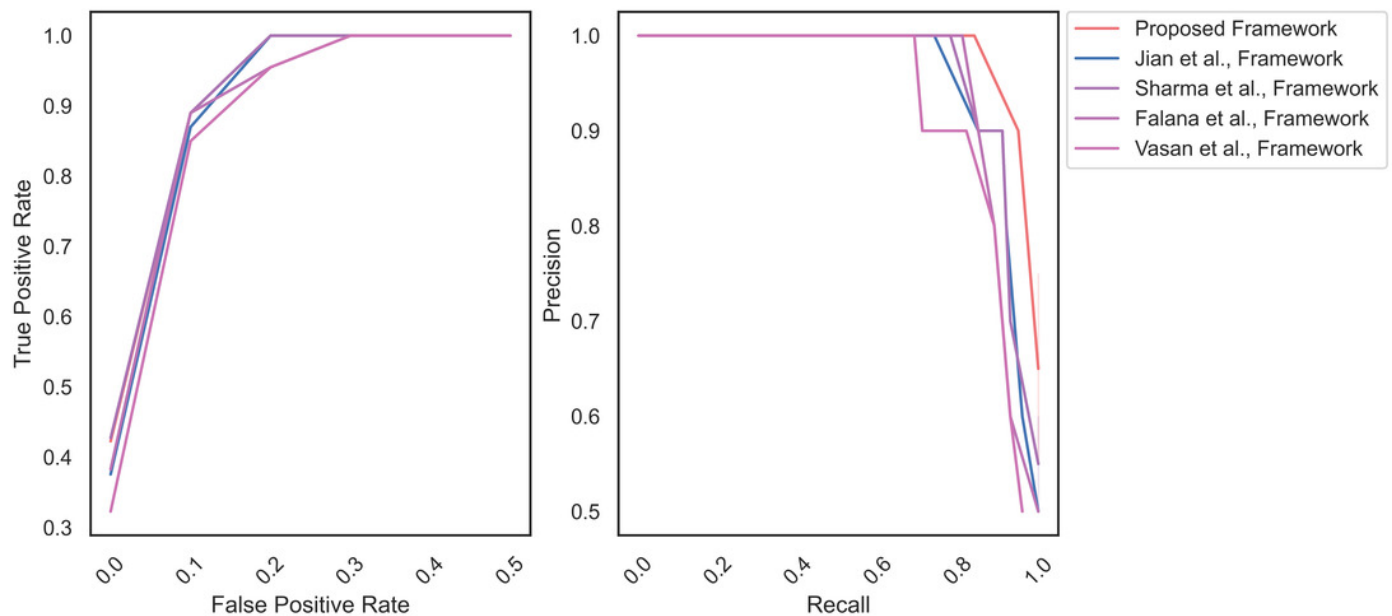
Figure 8

a_ IoT Malware Dataset b_ Virusshare Dataset